

21-25-00

A

04/24/00  
jc796 U.S. PTO

jc135 U.S. PTO  
09/557333  
04/24/00

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE**

In re Patent Application of: Mary Michelle Quinton et al.  
Title: SPEECH PROCESSING FOR TELEPHONY API  
Attorney Docket No.: 777.393US1

**PATENT APPLICATION TRANSMITTAL**

**BOX PATENT APPLICATION**

Assistant Commissioner for Patents  
Washington, D.C. 20231

We are transmitting herewith the following attached items and information (as indicated with an "X"):

**X CONTINUATION-IN-PART (CIP)** of prior Patent Application No. 09/157,469 (under 37 CFR § 1.53(b)) comprising:

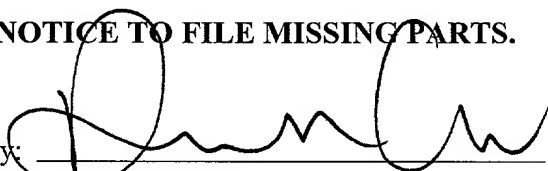
- X** Specification ( 39 pgs, including claims numbered 1 through 75 and a 1 page Abstract).
- X** Formal Drawing(s) ( 11 sheets).
- X** Unsigned Combined Declaration and Power of Attorney ( 4 pgs).
- X** Information Disclosure Statement ( 1 pgs), Form 1449 ( 1 pgs), and copies of cited references ( 1 ).
- X** Return postcard.

The filing fee (NOT ENCLOSED) will be calculated as follows:

	No. Filed	No. Extra	Rate	Fee
TOTAL CLAIMS	75 - 20 =	55	x 18 =	\$990.00
INDEPENDENT CLAIMS	15 - 3 =	12	x 78 =	\$936.00
1 MULTIPLE DEPENDENT CLAIMS PRESENTED				\$0.00
BASIC FEE				\$690.00
<b>TOTAL</b>				<b>\$2,616.00</b>

**THE FILING FEE WILL BE PAID UPON RECEIPT OF THE NOTICE TO FILE MISSING PARTS.**

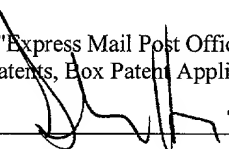
SCHWEGMAN, LUNDBERG, WOESSNER & KLUTH, P.A.  
P.O. Box 2938, Minneapolis, MN 55402 (612-373-6900)

By:   
Atty: D. C. Peter Chu  
Reg. No. 41,676

**Customer Number 21186**

"Express Mail" mailing label number: EL510128455US  
I hereby certify that this paper or fee is being deposited with the United States Postal Service "Express Mail Post Office to Addressee" service under 37 CFR 1.10 on the date indicated above and is addressed to the Assistant Commissioner for Patents, Box Patent Application, Washington, D.C. 20231.

By: Shawn L. Hise

Signature: 

UNITED STATES PATENT APPLICATION

**SPEECH PROCESSING FOR TELEPHONY API**

**INVENTORS**

**Mary Michelle Quinton**

**Stefan Solomon**

**Donald R. Ryan**

**Michael Clark**

Schwegman, Lundberg, Woessner, & Kluth, P.A.

1600 TCF Tower

121 South Eighth Street

Minneapolis, Minnesota 55402

ATTORNEY DOCKET 777.393US1

**MICROSOFT 113086.1**

## SPEECH PROCESSING FOR TELEPHONY API

5

### Related Applications

This application is a continuation-in-part of a U.S Serial No. 09/157,469, filed September 21, 1998.

10

### Technical Field

This invention relates generally to computer telephony, and more particularly to speech processing for computer telephony.

15

### Copyright Notice - Permission

A portion of the disclosure of this patent document contains material, which is subject to copyright protection. The copyright owner has no objection to the facsimile reproduction by anyone of the patent document or the patent disclosure as it appears in the Patent and Trademark Office patent files or records, but otherwise reserves all copyright rights whatsoever. The following notice applies to the software and data as described below and in the drawings attached hereto: Copyright © 1999, 2000, Microsoft Corporation, All Rights Reserved.

20

### Background

25

With the advent of computer networking, such as local-area networks (LAN), wide-area networks (WAN), intranets and the Internet, several applications have become popularized. In one such application, a user of a first client computer is able to "call" and communicate with a user of a second client computer. This type of application is generally known as computer telephony.

30

To accommodate computer telephony, operating systems such as versions of the MICROSOFT WINDOWS operating systems include telephony application

5 programming interfaces, or TAPI's. (It is noted that TAPI typically refers specifically to Microsoft's Telephony API and is not usually used in reference to other telephony API's. However, as used in this application, TAPI refers to telephony API's generically.) Application programming interfaces (API's) are interfaces by which computer programs can provide for specific functionality that  
10 is included within the operating systems. This means that programmers developing such programs do not have to develop their own code to provide this functionality, but rather can rely on the code within the operating system itself. Thus, a TAPI relates to a computer telephony application programming interface.

In the MICROSOFT WINDOWS 95 operating system, as well as other  
15 versions of the MICROSOFT WINDOWS operating system, TAPI version 2.1 provides for some basic computer telephony functionality for utilization by computer programs. In particular, TAPI 2.1 provides for call control -- the initiation and termination of computer telephony calls. However, call control is only one aspect of computer telephony. For example, once a computer telephony  
20 call is placed, the media aspects of the call must also be controlled. However, TAPI 2.1, as well as other prior art telephony API's, do not provide for this functionality.

The media aspects of the call relate to the information (or, media) that is itself the subject of the call. For example, a voice call includes audio information  
25 transmitted by both the caller and callee of a call, a video call includes both audio information and visual (video) information, etc. Currently, any multimedia devices that are to be used in conjunction with a computer telephony call -- such as microphones to detect sound, and speakers to play sound -- must have specific drivers written for this purpose, to be used specifically in conjunction with  
30 computer telephony calls. Other multimedia devices that may be present, in other words, may not be usable in conjunction with the call.

TAPI 2.1, as well as other prior art telephony API's, are also represented

5 as a framework that is not easily expanded. For example, TAPI 2.1 is  
procedurally based, which means the API cannot easily accommodate new aspects  
and features without redeveloping the entire API. For the reasons outlined in this  
background, as well as other reasons, there is, therefore, a need for the present  
invention.

10

### Summary

The above-identified problems, shortcomings and disadvantages with the  
prior art, as well as other problems, shortcoming and disadvantages, are solved by  
the present invention, which will be understood by reading and studying the  
15 specification and the drawings. In one embodiment, a system includes at least one  
call control object and at least one media control object. The call control objects  
are to initiate and terminate a computer telephony call having a media stream.  
The media control objects are to end-point the media stream of the computer  
telephony call. In a further embodiment, there is also a media control manager to  
20 instantiate a media control object for each multimedia device of the system.

Thus, embodiments of the invention provide for advantages not found in  
the prior art. The invention provides for well-defined media control: besides call  
control objects, embodiments of the invention include media control objects to  
end-point (for example, source or sink) the media stream of a computer telephony  
25 call. The invention provides for the utilization of multimedia devices (including  
virtual devices as well as physical devices) that may not have been installed  
specifically for telephony purposes, via the media control manager instantiating  
media control objects for such devices. Furthermore, the invention provides for  
an object-based hierarchy to TAPI's (e.g., via the call control objects and the  
30 media control objects), to maximize flexibility and further expansion of TAPI's  
based on the invention.

The invention includes systems, methods, computers, application

5 programming interfaces, and computer-readable media of varying scope. Besides  
the embodiments, advantages and aspects of the invention described here, the  
invention also includes other embodiments, advantages and aspects, as will  
become apparent by reading and studying the drawings and the following  
description.

10

### Brief Description of the Drawings

Figure 1 shows a diagram of the hardware and operating environment in  
conjunction with which aspects of the invention may be practiced.

15 Figure 2 shows a block diagram of an object hierarchy according to one  
aspect of the present invention.

Figure 3 shows a block diagram of an architecture according to one aspect  
of the present invention.

Figure 4(a) shows a method for placing an outgoing computer telephony  
call according to one aspect of the present invention.

20 Figure 4(b) shows a method for receiving an incoming computer telephony  
call according to one aspect of the present invention.

Figure 5 is a block diagram of a system according to one aspect of the  
present invention.

25 Figure 6 is a block diagram of a system according to one aspect of the  
present invention.

Figure 7 is a block diagram of a system according to one aspect of the  
present invention.

Figure 8 is an object diagram of a collection of objects according to one  
aspect of the present invention.

30 Figure 9 is a structural diagram of a data structure according to one aspect  
of the present invention.

Figure 10 is a process diagram of a method according to one aspect of the

5 present invention.

Figure 11 is a structural diagram of a data structure according to one aspect of the present invention.

Figure 12 is a process diagram of a method according to one aspect of the present invention.

10

### Detailed Description

In the following detailed description of exemplary embodiments of the invention, reference is made to the accompanying drawings which form a part hereof, and in which is shown, by way of illustration, specific exemplary embodiments in which the invention may be practiced. In the drawings, like  
15 numerals describe substantially similar components throughout the several views. These embodiments are described in sufficient detail to enable those skilled in the art to practice the invention. Other embodiments may be utilized and structural, logical, electrical, and other changes may be made without departing from the spirit or scope of the present invention. The following detailed description is,  
20 therefore, not to be taken in a limiting sense, and the scope of the present invention is defined only by the appended claims.

### Hardware and Operating Environment

Figure 1 is a block diagram of a system according to one aspect of the  
25 present invention. Figure 1 provides a brief, general description of a suitable computing environment in which the invention may be implemented. The invention will hereinafter be described in the general context of computer-executable program modules containing instructions executed by a personal computer (PC). Program modules include routines, programs, objects,  
30 components, data structures, etc., that perform particular tasks or implement particular abstract data types. Those skilled in the art will appreciate that the invention may be practiced with other computer-system configurations, including

5 hand-held devices, multiprocessor systems, microprocessor-based programmable  
consumer electronics, network PCs, minicomputers, mainframe computers, and  
the like, which have multimedia capabilities. The invention may also be practiced  
in distributed computing environments where tasks are performed by remote  
processing devices linked through a communications network. In a distributed  
10 computing environment, program modules may be located in both local and  
remote memory storage devices.

Figure 1 shows a general-purpose computing device in the form of a  
conventional personal computer 120, which includes processing unit 121, system  
memory 122, and system bus 123 that couples the system memory and other  
15 system components to processing unit 121. System bus 123 may be any of several  
types, including a memory bus or memory controller, a peripheral bus, and a local  
bus, and may use any of a variety of bus structures. System memory 122 includes  
read-only memory (ROM) 124 and random-access memory (RAM) 125. A basic  
input/output system (BIOS) 126, stored in ROM 124, contains the basic routines  
20 that transfer information between components of personal computer 120. BIOS  
126 also contains start-up routines for the system. Personal computer 120 further  
includes hard disk drive 127 for reading from and writing to a hard disk (not  
shown), magnetic disk drive 128 for reading from and writing to a removable  
magnetic disk 129, and optical disk drive 130 for reading from and writing to a  
25 removable optical disk 131 such as a CD-ROM or other optical medium. Hard  
disk drive 127, magnetic disk drive 128, and optical disk drive 130 are connected  
to system bus 123 by a hard-disk drive interface 132, a magnetic-disk drive  
interface 133, and an optical-drive interface 134, respectively. The drives and  
their associated computer-readable media provide nonvolatile storage of computer-  
30 readable instructions, data structures, program modules, and other data for  
personal computer 120. Although the exemplary environment described herein  
employs a hard disk, a removable magnetic disk 129 and a removable optical disk



5 131, those skilled in the art will appreciate that other types of computer-readable media which can store data accessible by a computer may also be used in the exemplary operating environment. Such media may include magnetic cassettes, flash-memory cards, digital versatile disks, Bernoulli cartridges, RAMs, ROMs, and the like.

10 Program modules may be stored on the hard disk, magnetic disk 129, optical disk 131, ROM 124, and RAM 125. Program modules may include operating system 135, one or more application programs 136, other program modules 137, and program data 138. A user may enter commands and information into personal computer 120 through input devices such as a keyboard  
15 140 and a pointing device 142. Other input devices (not shown) may include a microphone, joystick, game pad, satellite dish, scanner, or the like. These and other input devices are often connected to the processing unit 121 through a serial-port interface 146 coupled to system bus 123; but they may be connected through other interfaces not shown in Figure 1, such as a parallel port, a game port, or a  
20 universal serial bus (USB). A monitor 147 or other display device also connects to system bus 123 via an interface such as a video adapter 148. In addition to the monitor, personal computers typically include other peripheral output devices such as a sound adapter 156, speakers 157, and additional devices such as printers.

Personal computer 120 may operate in a networked environment using  
25 logical connections to one or more remote computers such as remote computer 149. Remote computer 149 may be another personal computer, a server, a router, a network PC, a peer device, or other common network node. It typically includes many or all of the components described above in connection with personal computer 120; however, only a storage device 150 is illustrated in Figure  
30 1. The logical connections depicted in Figure 1 include local-area network (LAN) 151 and a wide-area network (WAN) 152. Such networking environments are commonplace in offices, enterprise-wide computer networks, intranets and the

5 Internet.

When placed in a LAN networking environment, PC 120 connects to local network 151 through a network interface or adapter 153. When used in a WAN networking environment such as the Internet, PC 120 typically includes modem 154 or other means for establishing communications over network 152. Modem 10 154 may be internal or external to PC 120, and connects to system bus 123 via serial-port interface 146. In a networked environment, program modules, such as those comprising Microsoft® Word which are depicted as residing within PC 120 or portions thereof may be stored in remote storage device 150. Of course, the network connections shown are illustrative, and other means of establishing a 15 communications link between the computers may be substituted.

Software may be designed using many different methods, including object-oriented programming methods. C++ is one example of common object-oriented computer programming languages that provide the functionality associated with object-oriented programming. Object-oriented programming methods provide a 20 means to encapsulate data members (variables) and member functions (methods) that operate on that data into a single entity called a class. Object-oriented programming methods also provide a means to create new classes based on existing classes.

An object is an instance of a class. The data members of an object are 25 attributes that are stored inside the computer memory, and the methods are executable computer code that acts upon this data, along with potentially providing other services. The notion of an object is exploited in the present invention in that certain aspects of the invention are implemented as objects in one embodiment.

An interface is a group of related functions that are organized into a named 30 unit. Each interface may be uniquely identified by some identifier. Interfaces have no instantiation, that is, an interface is a definition only without the executable code needed to implement the methods which are specified by the

5 interface. An object may support an interface by providing executable code for  
the methods specified by the interface. The executable code supplied by the object  
must comply with the definitions specified by the interface. The object may also  
provide additional methods. Those skilled in the art will recognize that interfaces  
are not limited to use in or by an object-oriented programming environment.

10

### System

Figure 2 shows a block diagram of an object hierarchy according to one  
embodiment of the invention. Figure 3 shows a block diagram of an architecture  
according to one embodiment of the invention. In this section of the detailed  
15 description, a description of a computerized system according to an embodiment  
of the invention is provided. The description is provided by reference to Figure 2  
and Figure 3.

Referring first to Figure 2, an object hierarchy according to an  
embodiment of the invention is shown. The system includes a telephony  
20 application programming interface object (TAPI object) 200, an address object  
202, a terminal object 204, a call object 206, and a call-hub object 208. For each  
of objects 202, 204, 206 and 208, only a single object of each type is shown in  
Figure 2 for purposes of clarity; however, there can be in one embodiment of the  
invention multiple instantiations of each of these objects. Each of the objects 202,  
25 204, 206 and 208 may in one embodiment correspond to a specific means for  
performing functionality of the object.

The interface object 200 provides an interface by which computer  
programs can access the functionality provided by these other objects. This means  
that the computer programs themselves do not have to include code for this  
30 functionality, but instead can rely on the functionality provided by the objects  
themselves as already existing, and as interfaced to such programs via the  
interface object 200. Application programming interfaces within operating

5 systems such as versions of the MICROSOFT WINDOWS operating system are known within the art.

The address object 202 is a type of first-party call control object. A call control object is an object that provides for the initiation and termination of a computer telephony call having a media stream -- that is, the object provides for  
10 the connection and ending of a call. In particular, the address object 202 is an object over which a computer telephony call may be placed. That is, the address object 202 represents a line or device that can make or receive calls on it. In different embodiments of the invention, the object represents a modem attached to a PSTN (Public Switching Telephone Network) phone line, an ISDN (Integrated  
15 Services Digital Network) hardware card attached to an ISDN line, a DSL (Digital Subscriber Loop) modem attached to a PSTN phone line having DSL capability, and an IP (Internet Protocol) address that is able to make IP telephony calls. However, the invention is not limited to a particular representation. The address object 202 is a first-party call control object in that it relates to a party of the  
20 telephony call -- for example, the caller or callee of the telephony call -- as opposed to a third party not specifically of the telephony call.

The terminal object 204 is a type of media control object. A media control object is an object that end-points the media stream of a computer telephony call. The media stream of a computer telephony call is the information that actually  
25 makes up the call -- for example, audio information in the case of a voice call, audio and image (video) information in the case of a video call, etc. A media control object end-points the media stream in that it can be a sink object, which is a finishing end point such as speaker or a monitor where the media stream ends or is "sunk" after it has been communicated from one party to the call to another  
30 party to the call, or a source object, which is a beginning end point such as a microphone or a speaker where the media stream begins or is "sourced" such that it is then communicated from one party to the call to another party to the call.

5 The terminal object 204 can represent physical devices, such as the microphone or speakers' on a sound card, a video camera, and a phone, as well as more dynamic, virtual devices, such as a video window on the screen, a file to which the media stream is saved, and a DTMF (Dual Tone Multiple Frequency) detector.

The call object 206 is another type of first-party call control object. In  
10 particular, the call object 206 represents an end-point of the computer telephony call. For example, for a caller to callee direct call, there would be two call objects 206, a first object representing the first end point of the call, and a second object representing the second end point of the call. In a conference call, there would be more than two call objects 206, one object 206 for each participant (end point).

15 The call-hub object 208 is a third-party call control object. The call-hub object 208 relates the call objects 206 for a particular computer telephony call. In other words, it represents a telephony connection itself, and is basically a collection of call objects that are all related because they are on the same telephony connection. For example, one type of call-hub object 208 is a tracking  
20 object in a call center environment, to track the callers on a particular call, the duration of the phone call, etc. A third-party call control object is also able to initiate and terminate a phone call. However, the object is a third-party call control object in that it does not specifically relate to a particular party of the telephony call, but rather may encompass all the parties of the call (as well as  
25 information regarding the call).

Referring next to Figure 3, a block diagram of an architecture according to one embodiment of the invention is shown. The architecture includes a TAPI application 300, the TAPI 302, a telephony server 304, a telephony service  
30 provider 306, a media stream provider 308, and a terminal manager 310. The TAPI application 300 is a computer program that utilizes the functionality provided by the TAPI 302. That is, the TAPI application 300 is any type of computer program that utilizes the TAPI 302, through which the application is

5     able to access telephony call control and media control functionality provided by  
the TAPI 302.

10     The telephony server 304 and the telephony service provider 306 make up  
the call control aspects of the architecture of Figure 3. The telephony server 304  
keeps track of all telephony capabilities on a given computerized system; for  
example, such as that found within versions of the MICROSOFT WINDOWS NT  
operating system. The telephone service provider 306 is a component used to  
control a specific piece of telephony hardware. Although only one provider 306 is  
shown in Figure 3, the invention is not so limited; there can be many such  
providers installed.

15     The media stream provider 308 and the terminal manager 310 make up the  
media control aspects of the architecture of Figure 3. The media stream provider  
308 is an extension of the provider 306, and works together with the provider 306  
to implement call control (via the provider 306) and media control (via the  
provider 308). All call control requests proceed through the telephony server 304  
20     to the provider 306, and all media control requests proceed through to the  
provider 308. The media stream provider 308 is a component used to control a  
specific media stream (such as audio, video, etc.). Furthermore, there is a media  
stream provider 308 for each different media stream; although only one provider  
308 is shown in Figure 3, the invention is not so limited -- there can be many such  
25     providers installed.

30     The terminal manager 310 is a media control manager. It is a component  
that instantiates a medial control object for each installed multimedia device. That  
is, it is a component that allows telephony applications (such as application 300) to  
use any multimedia device installed within a telephony environment. When the  
manager 310 is initialized, it discovers all multimedia devices that it can use that  
are installed on a given computer, such as sound cards, video capture cards, as  
well as other multimedia hardware; the invention is not so limited. The manager

5 than creates a medial control object, such as a terminal object, for each of these devices.' The manager 310 also creates terminal objects or media control objects for other media sources or sink that do not necessarily correspond to hardware, but rather to virtual devices. These types of device represent media stream processing that is performed by the computer itself, rather than specific hardware.

10 For example, these types of terminals may include a video window, a speech recognition engine, and a file; the invention is not so limited.

The TAPI 302 in one embodiment has an interface that defines how the provider 308 communicates with the terminal manager 310. This interface allows any provider 308 (there may be more than one provider 308, although for

15 purposes of clarity only one is shown in Figure 3) to query the manager 310 for the devices that are represented as terminal or media control objects. The interface also allows the provider 308 to determine from the manager 310 how to include these devices within media streams that the provider 308 is to set up. Therefore, the manager 310 allows any provider 308 to access the same set of

20 terminal or media control objects, and use them with any telephony hardware.

Figure 4(a) shows a method for placing an outgoing computer telephony call according to an embodiment of the invention. Figure 4(b) shows a method for receiving an incoming computer telephony call according to an embodiment of the invention. In this section of the detailed description, exemplary methods

25 according to embodiments of the invention are presented. This description is provided in reference to Figures 4(a) through 4(b). These exemplary methods are desirably realized at least in part as one or more programs running on a computer -- that is, as a program executed from a computer-readable medium such as a memory by a processor of a computer. The programs are desirably storable on a

30 computer-readable medium such as a floppy disk or a CD-ROM, for distribution and installation and execution on another (suitably equipped) computer.

5           Thus, in one embodiment, a computer program is executed by a processor  
of a computer from a medium therefrom, where the program may include address  
objects, call objects, terminal objects, and call-hub objects, as described in the  
previous section of the detailed description. Each of these objects may in one  
embodiment also correspond to a specific means for performing the functionality  
10 of the object. In another embodiment, the computer program also includes a  
terminal manager, which detects a plurality of multimedia devices and instantiates  
a terminal object for each multimedia device detected, as has also been described  
in the previous section of the detailed description.

#### 15   Exemplary Methods

Referring now to Figure 4(a), a flowchart of a method for placing an  
outgoing computer telephony call, according to an embodiment of the invention, is  
shown. In 400, a TAPI object is instantiated by an application program so that the  
program is able to use the functionality provided by the TAPI. In 402, the TAPI  
20 object is initialized. For example, a terminal manager is run to instantiate  
terminal objects for physical and virtual multimedia devices, as has been described  
in the previous section of the detailed description.

In 404, the TAPI object is queried for an enumeration of the address  
objects available from the TAPI object. Each address object has certain telephony  
capabilities -- for example, one may relate to an ISDN line, another to a PSTN  
25 line, etc. Thus, in 406, each address object is queried to learn its telephony  
capabilities. The desired address object or objects are then selected, depending on  
the type of call desired (e.g., a regular voice call may go over a PSTN line, a  
video call may go over one or more ISDN lines, etc.).

30           In 408, a call object is instantiated from a desired address object or objects.  
The call object thus relates to the computer performing the method of Figure 4(a)  
as being the caller for a specific computer telephony call utilizing the desired



5 address object or objects. In 410, the desired address object or objects are queried  
for an enumeration of the terminal objects available from the address object or  
objects. For example, an address object relating to a PSTN line over which voice  
calls are placed may have a terminal object relating to a microphone and a  
terminal object relating to a sound card connected to a speaker. Depending on the  
10 type of call desired, then, in 412 at least one desired terminal object enumerated in  
410 is selected. Finally, in 414, the outgoing computer telephony call is  
connected (i.e., placed) over the desired address object or objects utilizing the  
desired terminal object or objects.

Thus, placing a computer telephony call according to the embodiment of  
15 the invention of Figure 4(a) involves determining the address objects that are  
available such that a call may be placed over them, and selecting a desired address  
object or objects. A call object is created for the specific call to be placed. The  
terminal objects that are available for the utilized address objects are then  
determined, and the desired terminal objects are selected. The call is then placed,  
20 such that the address objects represent the communication media over which the  
call is placed, and the terminal objects represent the multimedia devices that act as  
end points for the media stream communicated over the communication media.

Referring next to Figure 4(b), a flowchart of a method for receiving an  
incoming computer telephony call, according to an embodiment of the invention,  
25 is shown. In 450, a TAPI object is instantiated by an application program so that  
the program is able to use the functionality provided by the TAPI. In 452, the  
TAPI object is initialized. For example, a terminal manager is run to instantiate  
terminal objects for physical and virtual multimedia devices, as has been described  
in the previous section of the detailed description.

30 In 454, the TAPI object is queried for an enumeration of the address  
objects available from the TAPI object. Each address object has certain telephony  
capabilities -- for example, one may relate to an ISDN line, another to a PSTN

5 line, etc. Thus, in 456, each address object is queried to learn its telephony capabilities. The desired address object or objects are then selected, depending on the type of call that is desired to be listened for (e.g., a regular voice call may be received over a PSTN line, a video call may be received over one or more ISDN lines, etc.).

10 In 458, an event callback is instantiated and registered on the TAPI object. The event callback is a request by the application program performing the method of Figure 4(b) to have the TAPI object notify the application program when the desired event occurs -- in this case, when an incoming call is received. In 460, the desired address object or objects are also registered with the TAPI object. These  
15 are the address object or objects over which an incoming computer telephony call is to be listen for by the TAPI object, such that upon occurrence of such an event, the application program performing the method of Figure 4(b) is notified. Thus, in 462, a notification of an incoming computer telephony call from the TAPI object is received on the event callback. In 464, the incoming computer telephony  
20 call is connected (i.e., received) over the desired address object or objects.

As has been described, receiving a computer telephony call according to the embodiment of the invention of Figure 4(b) involves determining the address objects that are available such that a call may be received over them, and selecting a desired address object or objects. An event callback is created and registered, so  
25 that notification is received when a call arrives over the desired address object or objects. The call is then received (created), such that the address objects represent the communication media over which the call is received.

### Speech Processing

30 Figure 5 is a block diagram of a system according to one aspect of the present invention. A system 500 is an interactive voice response system that is used to collect and provide information for a caller. When a caller calls the

5 system 500, the call is answered automatically. The system 500 presents the caller with a number of choices that the caller can select by pressing the keypad of the caller's telephone or by voice. If the system 500 determines that it is necessary for the caller to communicate to a human agent, the call is then routed to the client computer of the human agent so that the human agent can answer.

10 The system 500 includes a telephony source 502. The telephony source 502 generates a telephony call that is transmitted by a public switched telephone network 504. The public switched telephone network 504 transmits the telephony call to a gateway 506. The gateway 506 translates the telephony call based on the communication protocols of the public switched telephone network 504 to a  
15 telephony call based on internet protocols. The gateway 506 transmits the internet protocol telephony call to a call router 510. The call router 510 may store information associated with the internet protocol telephony call in a data store 514.

The call router 510 routes the internet protocol telephony call to an interactive voice response server 512. In one embodiment, the interactive voice  
20 response server 512 includes a terminal object. In another embodiment, the interactive voice response server 512 performs media processing tasks, such as playing prerecorded messages and detecting input from the user. In one embodiment, such media processing tasks can be accomplished using an appropriate instantiation of the terminal object. The interactive voice response  
25 server 512 may store information associated with the internet protocol telephony call in the data store 514. The interactive voice response server 512 decides to allow the call router 510 to route the internet telephony call to a client computer 516 depending on a caller's interaction with the interactive voice response server 512.

30 The client computer 516 is adapted to retrieve information associated with the internet protocol telephony call from the data store 514. In one embodiment, the client computer 516 includes a terminal object. The terminal object allows the

5 client computer to answer the internet protocol telephony call.

• Figure 6 is a block diagram of a system according to one aspect of the present invention. A system 600 is a unified messaging system that allows voice mail to be saved as a computer file so that the voice mail can be accessed through an email system.

10 The system 600 includes a telephony source 602. The telephony source 602 generates a telephony call that is transmitted to a gateway 604. The gateway 604 translates the telephony call to a telephony call based on internet protocols. The gateway 604 transmits the internet protocol telephony call to a client computer 610. If the client computer 610 is unavailable to answer the internet protocol  
15 telephony call, the internet protocol telephony call is routed to a voice mail system 606. In one embodiment, the voice mail system 606 includes a terminal object.

The voice mail system 606 saves the voice mail in the email store. The client computer 610 receives an email message with the voice mail saved as an attachment. The client computer 610 may then access the voice mail through the  
20 media processing capability of the client computer 610.

Figure 7 is a block diagram of a system according to one aspect of the present invention. A system 700 illustrates speech-enabled Web applications. The system 700 allows Web content and services to be accessed through telephony connections and rendered as speech rather than as text or graphics.

25 The system 700 includes a client 702. The client 702 initiates a telephony call to a voice browser 704 that is executing on a computer. The voice browser 704 can access at least one Web page 708 stored on a Web server 710. The Web page 708 may include voice tags. In one embodiment, the voice browser 704 includes a terminal object that can interpret the voice tags. In another  
30 embodiment, the terminal object renders the Web page 708 into speech for the client 702. In another embodiment, the terminal object allows the client to navigate through a Web site based on the speech commands of the client.

5           Figure 8 is an object diagram of a collection of objects according to one aspect of the present invention. An object hierarchy 801 includes objects that are similar to objects discussed hereinbefore. For clarity purposes, discussion relating to those similar objects is incorporated in full here.

10           The object hierarchy 801 includes a TAPI object 800, a call-hub object 802, an address object 804, a call object 806, a terminal object 808, and a stream object 810. These objects have been discussed hereinbefore. The object hierarchy 801 includes a speech recognition object 812 that is derived from the terminal object 808. In one embodiment, the terminal object 808 can be viewed as a terminal data structure, and the speech recognition object 812 is a speech  
15           recognition data structure that extends the terminal data structure. The object hierarchy 801 also includes a speech generation object 814 that is derived from the terminal object 808. In one embodiment, the terminal object 808 can be viewed as a terminal data structure, and the speech recognition object 814 is a speech recognition data structure that extends the terminal data structure.

20           Figure 9 is a structural diagram of a data structure according to one aspect of the present invention. A data structure 900 supports speech recognition. The data structure 900 includes a number of data structures to help the process of speech recognition. These data structures include an engine token data structure 902, an enumeration engine data structure 908, a speech recognition data structure 912, and a recognition context data structure 922.  
25

          The engine token data structure 902 includes a method member get engine name 904. The method member get engine name 904 gets the name of a speech recognition engine in a textual form. The engine token data structure 902 includes a method member get engine token 906. The method member get engine token  
30           906 gets an identifier that identifies a speech recognition engine.

5           The enumeration engine data structure 908 includes a method member next 910. The method member next 910 gets the next available speech recognition engine from a list of available speech recognition engines.

          The speech recognition data structure 912 includes a method member enumerate recognition engines 914. The method member enumerate recognition engines 914 obtains an indirect reference to a listing of speech recognition engines that are available for use. The speech recognition data structure 912 includes a method member select engine 916. The method member select engine 916 selects a speech recognition engine to be used in the speech recognition process. The speech recognition data structure 912 includes a method member get selected engine 918. The method member get selected engine 918 retrieves the currently selected speech recognition engine. The speech recognition data structure 912 includes a method member convert extended markup language to grammar 920 converts extended markup language (XML) into a compiled grammar for use with a speech recognition engine.

20           The recognition context 922 includes a method member initialize 924. The method member initialize 924 creates a speech recognition context based on a selected speech recognition engine. The recognition context 922 includes a method member shut down 926. The method member shut down 926 destroys a speech recognition context. The recognition context 922 includes a method member load grammar 928. The method member load grammar 928 loads a grammar into a recognition context from a source selected from a group consisting of a resource, a memory, and a file. The recognition context 922 includes a method member unload grammar 930. The method member unload grammar 930 unloads a grammar previously loaded into a recognition context. The recognition context 30 922 includes a method member activate grammar 932. The method member activate grammar 932 activates a grammar to be used in a speech recognition engine. The recognition context 922 includes a method member get result 934.

5 The method member get result 934 retrieves a speech recognition result. The recognition context 922 includes a method member get hypothesis 936. The method member get hypothesis 936 retrieves a speech recognition result that is deemed a likely speech recognition result.

Figure 10 is a process diagram of a method according to one aspect of the present invention. A process 1000 is a method for enhancing media processing. The process 1000 includes an act 1002 for selecting a speech recognition terminal object. The process 1000 includes an act 1004 for requesting a speech recognition terminal object.

The process 1000 includes an act 1006 for getting a desired speech recognition engine. The act 1006 includes an act for enumerating a list of available speech recognition engines, an act for identifying a desired speech recognition engine from the list of available speech recognition engines, and an act for selecting the desired speech recognition engine.

The process 1000 includes an act 1008 for setting a speech recognition context. The act 1008 includes an act for initializing the speech recognition context, an act for loading a grammar for the speech recognition context, and an act for setting the speech recognition context to notify a user when a desired event occurs.

Figure 11 is a structural diagram of a data structure according to one aspect of the present invention. A data structure 1100 supports speech generation. The data structure 1100 takes a text string and renders such a text string into speech. The data structure 1100 allows a voice to be selected to speak the rendered speech.

The data structure 1100 includes a set of voice method members that are selected from a group consisting of a method member set voice 1102 for setting a voice to be used for speech generation and a method member get voice 1104 for getting the voice used in speech generation. The data structure 1100 includes a set of priority method members that are selected from a group consisting of a method

5 member set priority 1106 for setting a priority for a voice and a method member  
get priority 1108 for getting a priority for a voice. The voice with a higher  
priority may interrupt a voice with a lower priority. The data structure 1100  
includes a set of volume method members that are selected from a group  
consisting of a method member set volume 1110 for setting a volume of speech  
10 synthesized by a speech engine and a method member get volume 1112 for getting  
a volume of speech synthesized by a speech generation engine. The data structure  
1100 includes a set of rate method members that are selected from a group  
consisting of a method member set rate 1114 for setting a rate of speech  
synthesized by a speech generation engine and a method member get rate 1116 for  
15 getting a rate of speech synthesized by a speech generation engine. The data  
structure 1100 includes a set of time out method members that are selected from a  
group consisting of a method member set time 1118 for setting a time for a speech  
synthesis to time out and a method member get time 1120 for getting a time for a  
speech synthesis to time out.

20 The data structure 1100 includes a method member speak 1128 for  
synthesizing text to audio. The data structure 1100 includes a method member get  
status 1122 for getting a status on synthesizing of output audio. The data structure  
1100 includes a method member skip 1124 for skipping to a specific point in a text  
stream. The data structure 1100 includes a method member wait 1126 for blocking  
25 other executions until the method member speak 1128 has been executed to  
completion. The data structure 1100 includes a method member enumerate voices  
1130 for obtaining a list of voices for the speech generation engine.

The method member speak 1128 is receptive to a number of inputs so as to  
enhance the synthesis of text to audio. These inputs include a text stream with  
30 voice markup, an offset that represents an offset into the text stream where the  
voice should start speaking, a speakover flag so as to blend the voice output over



5 any currently playing audio output, and a punctuation flag so as to allow a speech generation engine to speak each punctuation of a text stream.

Figure 12 is a process diagram of a method according to one aspect of the present invention. A process 1200 is a method for enhancing media processing. The process 1200 allows speech generation. The process 1200 includes an act  
10 1202 for requesting a speech generation terminal object. The process 1200 includes an act 1204 for selecting a voice. The act 1204 includes an act for enumerating a list of available voices and an act for identifying a desired voice from the list of available voices. The process 1200 includes an act 1206 for generating a speech. In one embodiment, the act 1206 generates the speech from a  
15 text stream that includes voice markup.

### Conclusion

Computer telephony application programming interface has been described.

Although the specific embodiments have been illustrated and described  
20 herein, it will be appreciated by those of ordinary skill in the art that any arrangement which is calculated to achieve the same purpose may be substituted for the specific embodiments shown. This application is intended to cover any adaptations or variations of the present invention. It is to be understood that the above description is intended to be illustrative, and not restrictive. Combinations  
25 of the above embodiments and other embodiments will be apparent to those of skill in the art upon reviewing the above description. The scope of the invention includes any other applications in which the above structures and fabrication methods are used. Accordingly, the scope of the invention should only be determined with reference to the appended claims, along with the full scope of  
30 equivalents to which such claims are entitled.

5 We claim:

1. An enhanced interactive voice response system, comprising:

a call router to route an internet protocol telephony call; and

an interactive voice response server to receive the internet protocol

10 telephony call from the call router, wherein the interactive voice response server includes a terminal object.

2. The system of claim 1, further comprising a gateway coupled to the call router.

15

3. The system of claim 2, further comprising a public switched telephone network coupled to the gateway.

4. The system of claim 2, wherein the gateway translates telephony calls

20 based on communication protocols of a public switched telephone network to telephony calls based on internet protocols.

5. The system of claim 1, further comprising a client computer, wherein the client computer includes a terminal object so as to receive the internet telephony

25 call routed from the router.

5

6. An enhanced interactive voice response system, comprising:

a data store;

a call router to route an internet protocol telephony call; and

an interactive voice response server to receive the internet protocol

10 telephony call from the call router, wherein the interactive voice response server includes a terminal object.

7. The system of claim 6, wherein the call router stores call information in the data store.

15

8. The system of claim 6, wherein the interactive voice response server stores call information in the data store.

9. The system of claim 6, further comprising a client computer, wherein the  
20 client computer includes a terminal object so as to receive the internet telephony call routed from the router.

10. The system of claim 9, wherein the client computer is adapted to retrieve call information from the data store.

25

5 11. An enhanced unified message system, comprising:  
an email store; and  
a voice mail system to receive an internet protocol telephony call, wherein  
the voice mail system includes a terminal object.

10 12. The system of claim 11, further comprising a gateway to transmit an  
internet protocol telephony call.

13. The system of claim 12, further comprising a client computer to receive the  
internet protocol telephony call from the gateway.

15

14. The system of claim 13, wherein the voice mail system saves the internet  
protocol telephony call in the email store.

15. The system of claim 14, wherein the client computer is adapted to access a  
20 saved internet protocol telephony call through the email store.

16. A system to enhance speech-enabled Web applications, comprising:  
a Web page that includes voice tags; and  
a voice browser that includes a terminal object to interpret the voice tags.

25

5 17. The system of claim 16, further comprising a Web server that stores the Web page.

18. The system of claim 16, further comprising a client that couples to the voice browser through a telephone call.

10

19. The system of claim 18, wherein the terminal object of the voice browser renders the Web page into speech for the client that couples to the voice browser through the telephone call.

15 20. The system of claim 18, wherein the terminal object of the voice browser allows the client to navigate through a Web site based on the speech commands of the client.

21. A data structure to enhance media processing, comprising:  
20 a terminal data structure to instantiate terminal objects; and  
a speech recognition terminal data structure that extends the terminal data structure.

22. The data structure of claim 21, wherein the speech recognition terminal  
25 data structure includes an engine token data structure.

23. The data structure of claim 21, wherein the speech recognition terminal data structure includes an enumeration engine data structure.

24. The data structure of claim 21, wherein the speech recognition terminal data structure includes a speech recognition data structure.

25. The data structure of claim 21, wherein the speech recognition terminal data structure includes a recognition context data structure.

26. A data structure to enhance media processing, comprising:  
a terminal data structure to instantiate terminal objects; and  
a speech recognition terminal data structure that extends the terminal data structure, wherein the speech recognition terminal data structure includes an engine token data structure.

20

27. The data structure of claim 26, wherein the engine token data structure includes a method member get engine name for getting a name of a speech recognition engine in a textual form.

28. The data structure of claim 26, wherein the engine token data structure

5 includes a method member get engine token for getting an identifier that identifies  
a speech recognition engine.

29. A data structure to enhance media processing, comprising:  
a terminal data structure to instantiate terminal objects; and  
10 a speech recognition terminal data structure that extends the terminal data  
structure, wherein the speech recognition terminal data structure includes an  
enumeration engine data structure.

30. The data structure of claim 29, wherein the enumeration engine data  
15 structure includes a method member next for getting a next available speech  
recognition engine.

31. A data structure to enhance media processing, comprising:  
a terminal data structure to instantiate terminal objects; and  
20 a speech recognition terminal data structure that extends the terminal data  
structure, wherein the speech recognition terminal data structure includes a speech  
recognition data structure.

32. The data structure of claim 31, wherein the speech recognition data  
25 structure includes a member method enumerate recognition engines for obtaining

5 an indirect reference to a listing of speech recognition engines that are available  
for use.

33. The data structure of claim 31, wherein the speech recognition data  
structure includes a member method select engine for selecting a speech  
10 recognition engine to be used.

34. The data structure of claim 31, wherein the speech recognition data  
structure includes a member method get selected engine for retrieving the  
currently selected speech recognition engine.

15 35. The data structure of claim 31, wherein the speech recognition data  
structure includes a member method convert extended markup language to  
grammar for converting extended markup language text into a compiled grammar  
for use with a speech recognition engine.

20 36. A data structure to enhance media processing, comprising:  
a terminal data structure to instantiate terminal objects; and  
a speech recognition terminal data structure that extends the terminal data  
structure, wherein the speech recognition terminal data structure includes a  
25 recognition context data structure.



37. The data structure of claim 36, wherein the recognition context data structure includes a method member initialize for creating a speech recognition context based on a selected speech recognition engine.

10 38. The data structure of claim 36, wherein the recognition context data structure includes a method member shut down for destroying a speech recognition context.

39. The data structure of claim 36, wherein the recognition context data  
15 structure includes a method member load grammar for loading a grammar into a recognition context from a source selected from a group consisting of a resource, a memory, and a file.

40. The data structure of claim 36, wherein the recognition context data  
20 structure includes a method member unload grammar for unloading a grammar previously loaded into a recognition context.

41. The data structure of claim 36, wherein the recognition context data structure includes a method member activate grammar for activating a grammar to  
25 be used in a speech recognition engine.

42. The data structure of claim 36, wherein the recognition context data structure includes a method member get result for retrieving a speech recognition result.

10 43. The data structure of claim 36, wherein the recognition context data structure includes a method member get hypothesis for retrieving a speech recognition result that is deemed a likely speech recognition result.

44. A method for enhancing media processing, comprising:  
15 requesting a speech recognition terminal object;  
getting a desired speech recognition engine; and  
setting a speech recognition context.

45. The method of claim 44, further comprising selecting a speech recognition  
20 terminal object.

46. The method of claim 44, wherein getting includes enumerating a list of available speech recognition engines.

25 47. The method of claim 46, wherein getting includes identifying a desired

5 speech recognition engine from the list of available speech recognition engines.

48. The method of claim 47, wherein getting includes selecting the desired speech recognition engine.

10 49. The method of claim 44, wherein setting includes initializing the speech recognition context.

50. The method of claim 44, wherein setting includes loading a grammar for the speech recognition context.

15

51. The method of claim 44, wherein setting includes activating a grammar for the speech recognition context.

52. The method of claim 44, wherein setting includes setting the speech  
20 recognition context to notify a user when a desired event occurs.

53. A computer readable medium having instructions stored thereon for causing a computer to perform a method for enhancing media processing, the method comprising:

25 requesting a speech recognition terminal object;

5           getting a desired speech recognition engine; and  
          setting a speech recognition context.

54.     A data structure to enhance media processing, comprising:  
          a terminal data structure to instantiate terminal objects; and  
10        a speech generation terminal data structure that extends the terminal data  
          structure.

55.     The data structure of claim 54, wherein the speech generation terminal data  
          structure includes voice method members that are selected from a group consisting  
15        of a method member set voice for setting a voice to be used for speech generation  
          and a method member get voice for getting the voice used in speech generation.

56.     The data structure of claim 54, wherein the speech generation terminal data  
          structure includes priority method members that are selected from a group  
20        consisting of a method member set priority for setting a priority for a voice and a  
          method member get priority for getting a priority for a voice, wherein a voice with  
          a higher priority may interrupt a voice with a lower priority.

57.     The data structure of claim 54, wherein the speech generation terminal data  
25        structure includes volume method members that are selected from a group

5       consisting of a method member set volume for setting a volume of speech  
synthesized by a speech generation engine and a method member get volume for  
getting a volume of speech synthesized by a speech generation engine.

58.       The data structure of claim 54, wherein the speech generation terminal data  
10       structure includes rate method members that are selected from a group consisting  
of a method member set rate for setting a rate of speech synthesized by a speech  
generation engine and a method member get rate for getting a rate of speech  
synthesized by a speech generation engine.

15       59.       The data structure of claim 54, wherein the speech generation terminal data  
structure includes time out method members that are selected from a group  
consisting of a method member set time for setting a time for a speech synthesis to  
time out and a method member get time for getting a time for a speech synthesis to  
time out.

20       60.       The data structure of claim 54, wherein the speech generation terminal data  
structure includes a method member speak for synthesizing text to audio.

61.       The data structure of claim 54, wherein the speech generation terminal data  
25       structure includes a method member get status for getting a status on synthesizing

5 of output audio.

62. The data structure of claim 54, wherein the speech generation terminal data structure includes a method member skip for skipping to a specific point in a text stream.

10

63. The data structure of claim 60, wherein the speech generation terminal data structure includes a method member wait for blocking other executions until the method member speak has been executed to completion.

15 64. The data structure of claim 60, wherein the speech generation terminal data structure includes a method member enumerate voices for obtaining a list of voices for the speech generation engine.

20 65. A data structure to enhance media processing, comprising:  
a terminal data structure to instantiate terminal objects; and  
a speech generation terminal data structure that extends the terminal data structure, wherein the speech generation terminal data structure includes a method member speak for synthesizing text to audio.

25 66. The data structure of claim 65, wherein the method member speak is

5 receptive to a text stream with voice markup to be synthesized.

67. The data structure of claim 65, wherein the method member speak is  
receptive to an offset that represents an offset into a text stream where the voice  
should start speaking.

10

68. The data structure of claim 65, wherein the method member speak is  
receptive to a speakover flag so as to blend the voice output over any currently  
playing audio output.

15 69. The data structure of claim 65, wherein the method member speak is  
receptive to a punctuation flag so as to allow a speech generation engine to speak  
each punctuation of a text stream.

20 70. A method for enhancing media processing, comprising:  
requesting a speech generation terminal object; and  
generating a speech.

71. The method of claim 70, wherein generating includes generating the speech  
from a text stream that includes voice markup.

25

5     72.     The method of claim 70, further comprising selecting a voice.

73.     The method of claim 72, wherein selecting includes enumerating a list of  
available voices.

10    74.     The method of claim 73, wherein selecting includes identifying a desired  
voice from the list of available voices.

75.     A computer readable medium having instructions stored thereon for  
causing a computer to perform a method for enhancing media processing, the

15    method comprising:

requesting a speech generation terminal object; and

generating a speech.



Abstract of the Disclosure

Systems, methods, and structures are discussed that enhance media processing. One aspect of the present invention includes a data structure to enhance media processing. The data structure includes a terminal data structure to instantiate terminal objects and a speech recognition terminal data structure that extends the terminal data structure. Another aspect of the present invention includes a data structure to enhance media processing. This data structure includes a terminal data structure to instantiate terminal objects and a speech generation terminal data structure that extends the terminal data structure. These data structures may be used to implement an internet protocol interactive voice response system, an internet protocol unified message system, and speech-enabled Web applications.

"Express Mail" mailing label number: EL510 128455US  
Date of Deposit: Apr 1 04 2000  
I hereby certify that this paper or fee is being deposited with the  
United States Postal Service "Express Mail Post Office to Addressee"  
service under 37 CFR 1.10 on the date indicated above and is  
addressed to the Assistant Commissioner for Patents,  
Washington, D.C. 20231  
Printed Name Shirley L. Niss  
Signature [Signature]

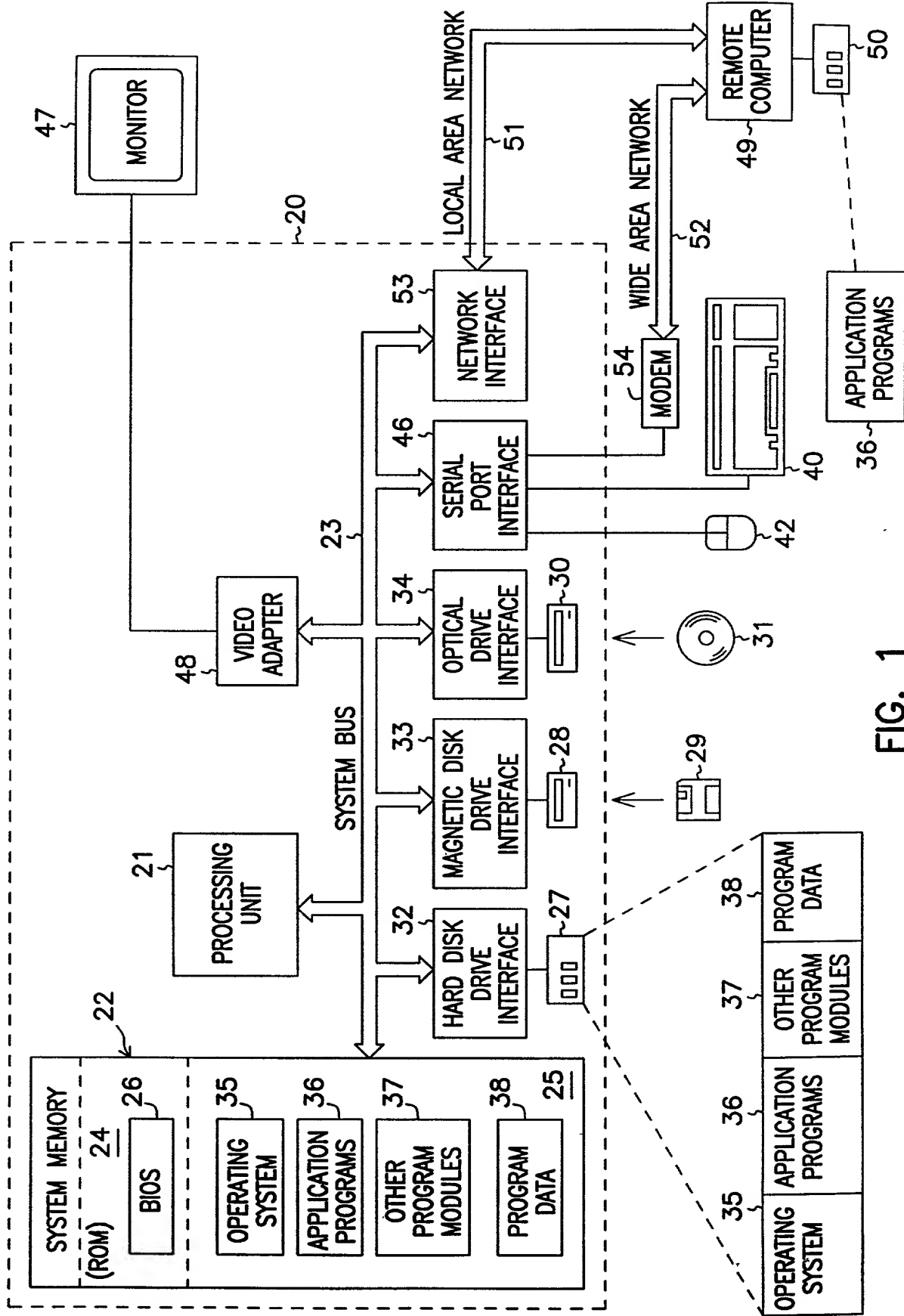


FIG. 1

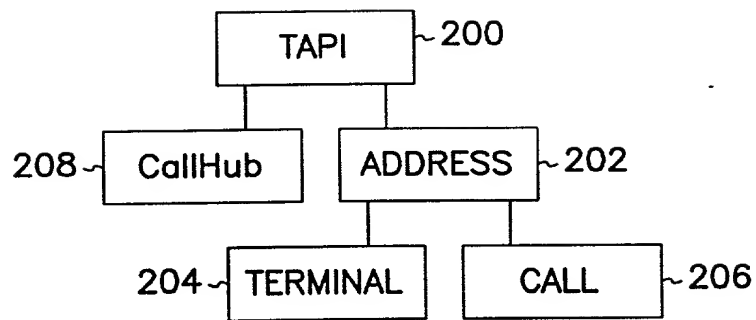


FIG. 2

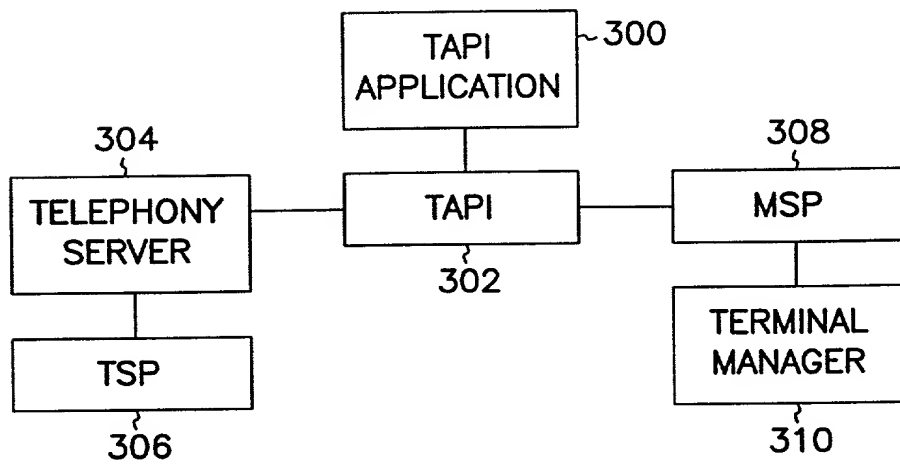


FIG. 3

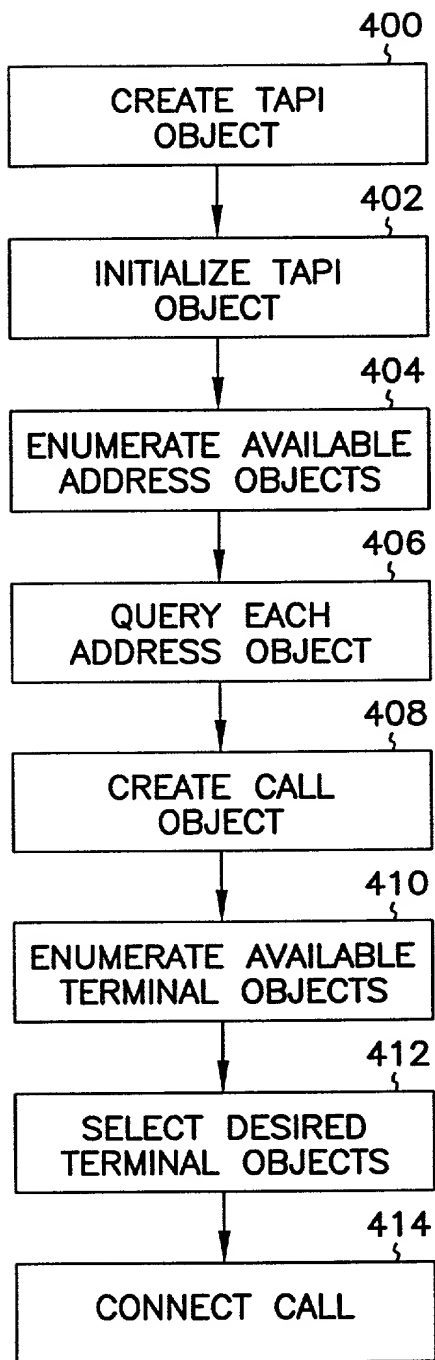


FIG. 4(a)

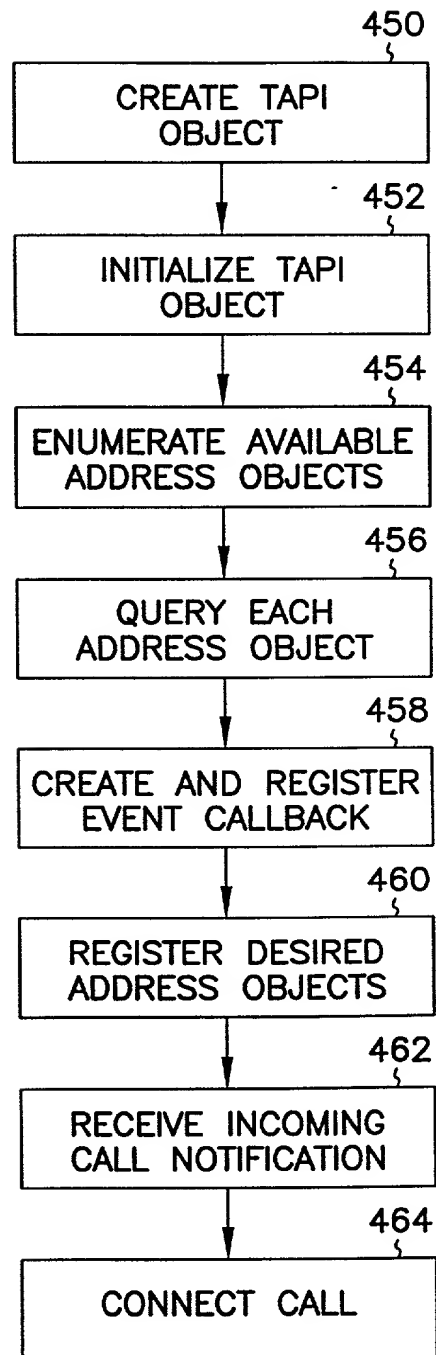


FIG. 4(b)

FIG. 5 is a block diagram of a network architecture.

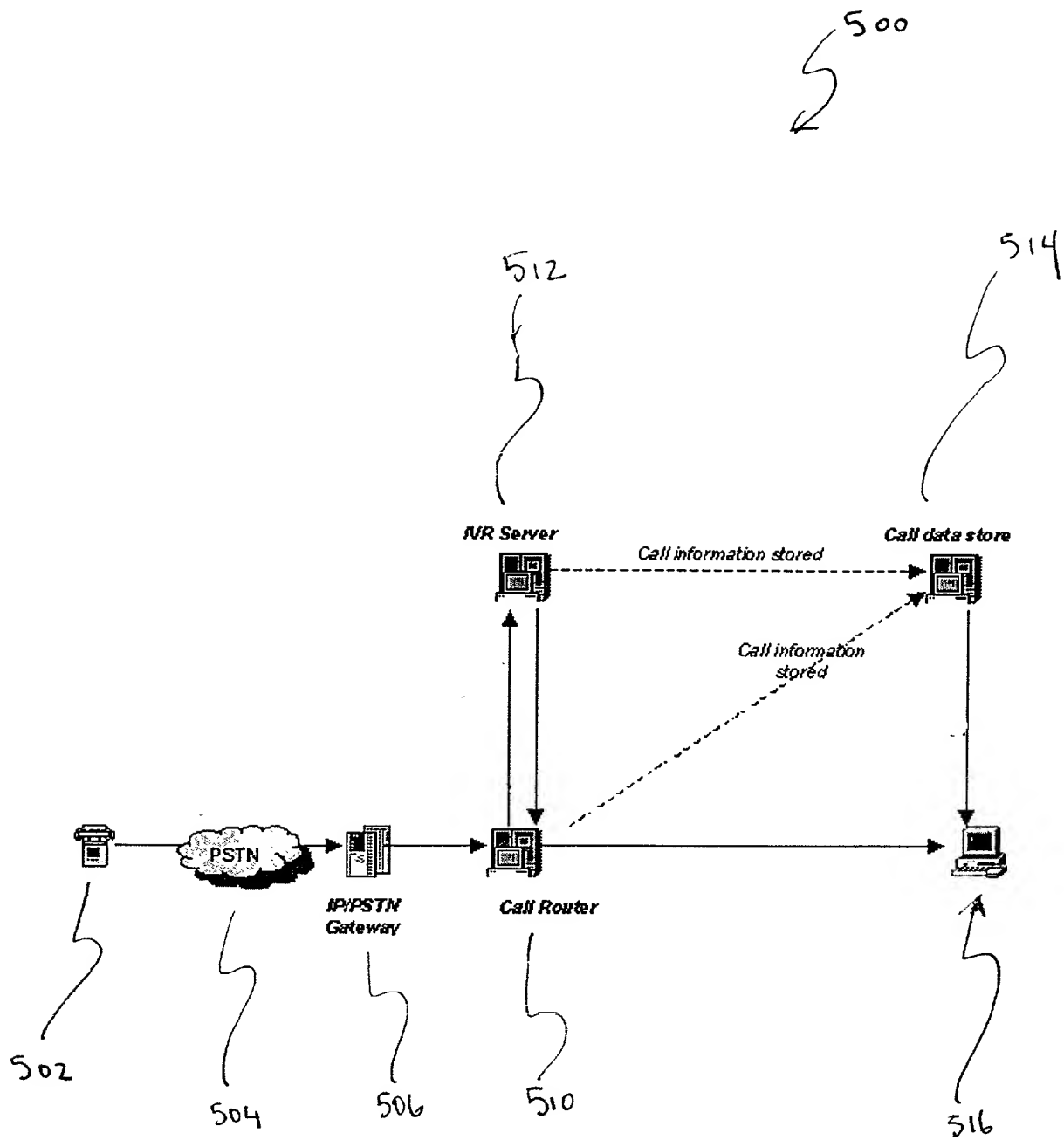


FIG. 5

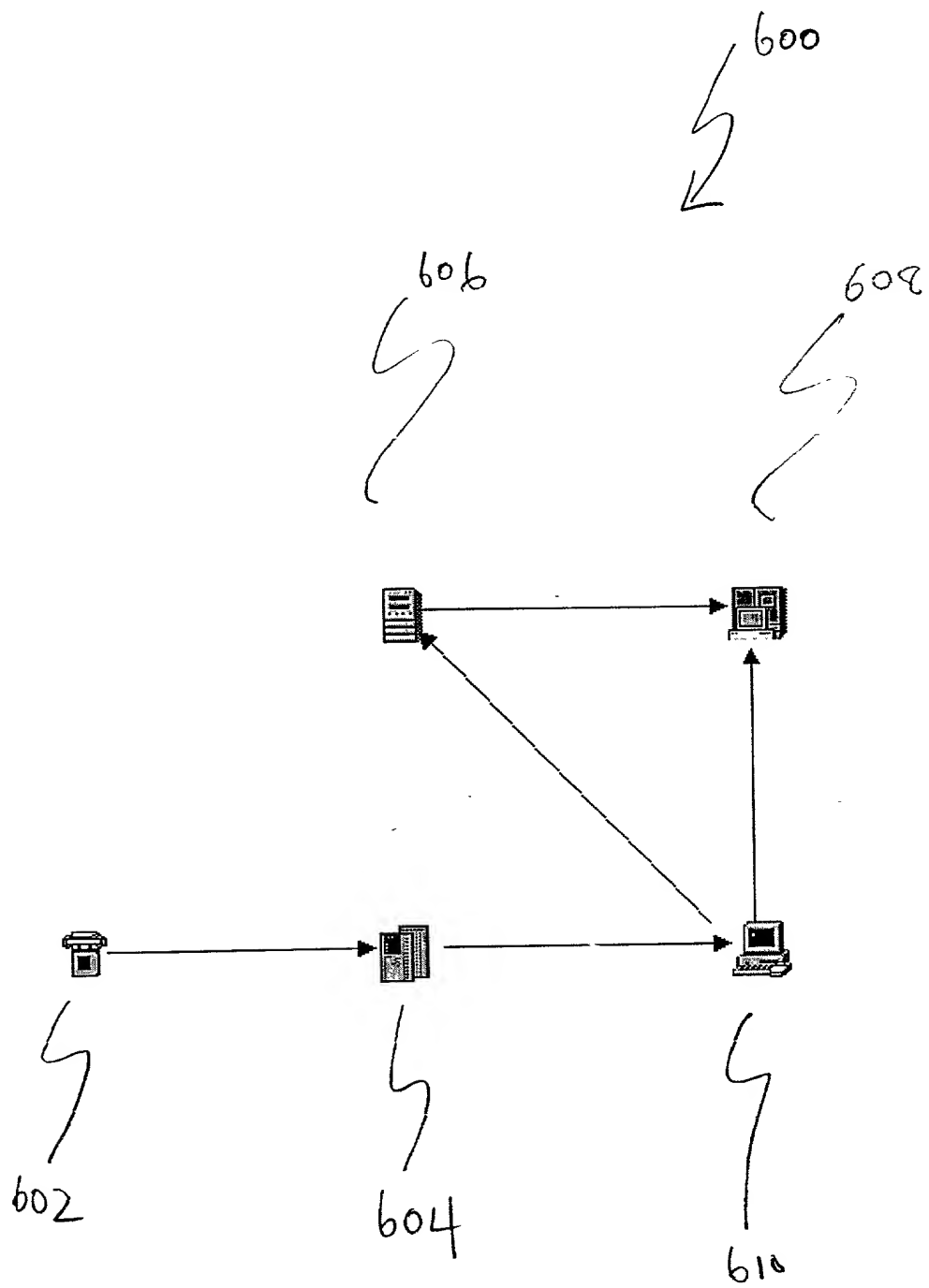


Fig. 6

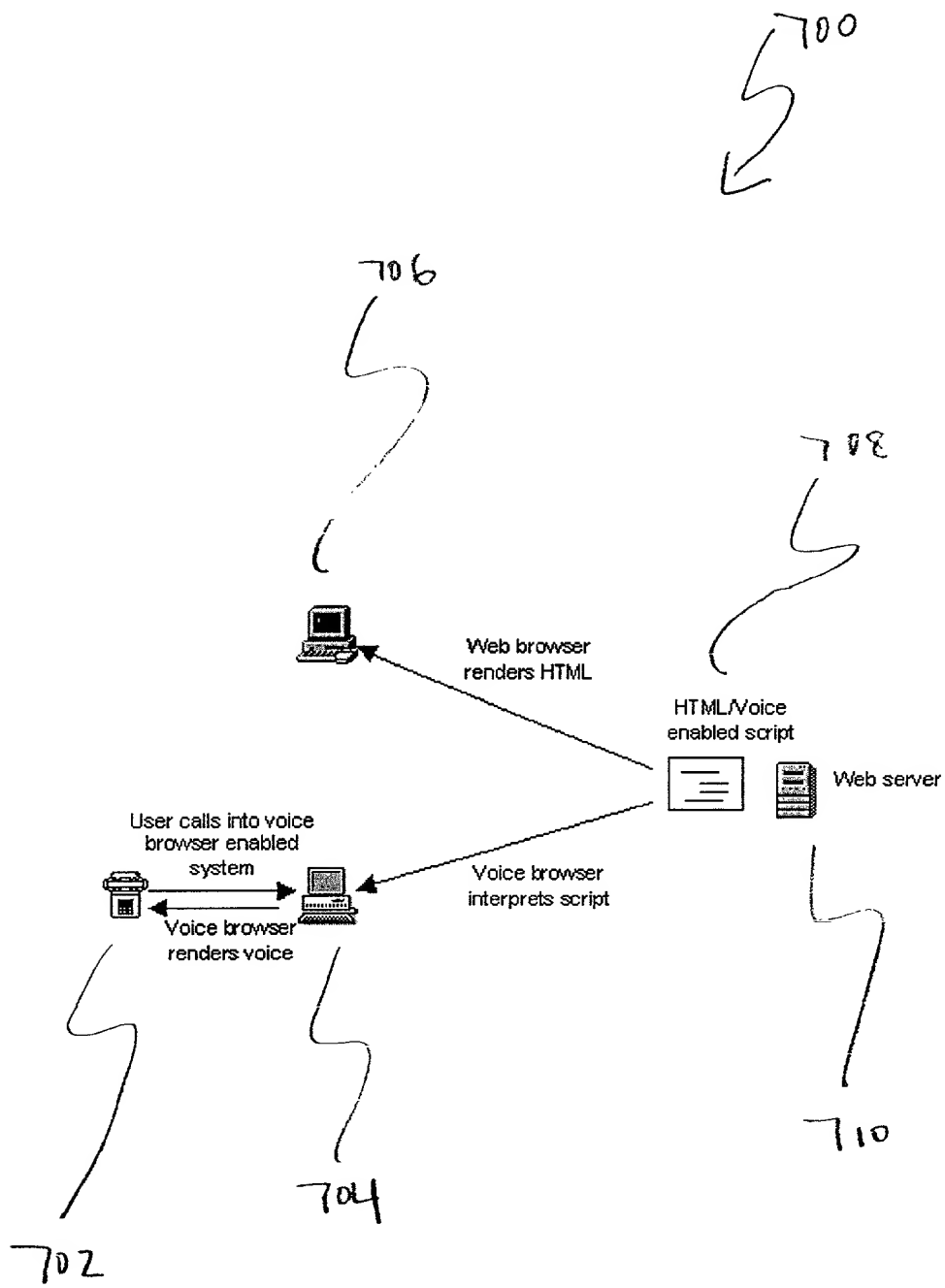


FIG. 7

FIG. 8 is a block diagram of a system for processing a call.

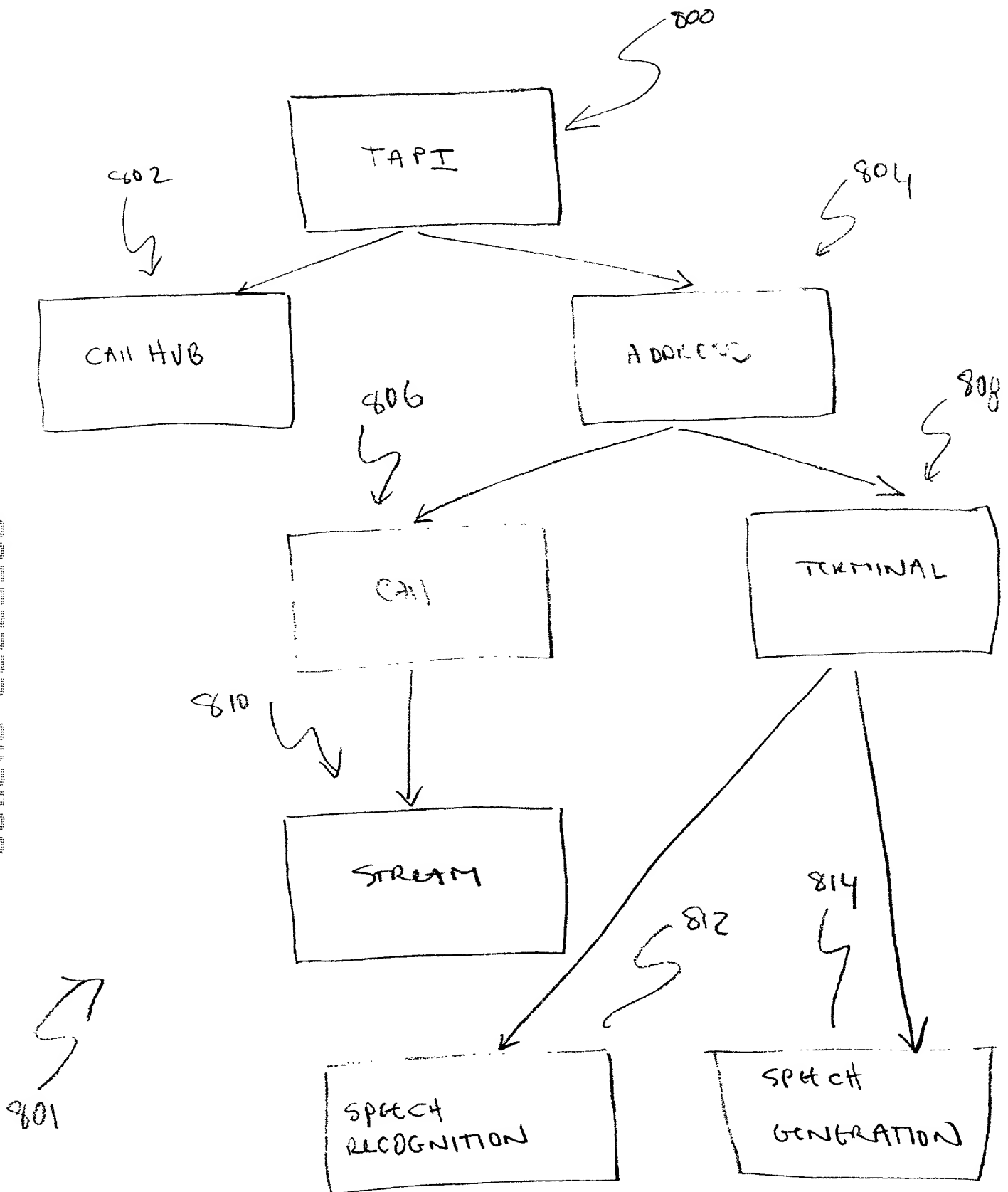


FIG 8



# SPEECH RECOGNITION TERMINAL

ENGINE TOKEN

GET ENGINE NAME()

GET ENGINE TOKEN()

ENUMERATION ENGINE

NEXT()

902

904

906

908

912

914

916

918

920

922

910

924

926

928

930

932

934

936

SPEECH RECOGNITION

ENUMERATE RECOGNITION ENGINES()

SELECT ENGINE()

GET SELECTED ENGINE()

CONVERT EXTENDED MARKUP LANGUAGE TO GRAMMAR()

RECOGNITION CONTEXT

INITIALIZE()

SHUTDOWN()

LOAD GRAMMAR()

UNLOAD GRAMMAR()

ACTIVATE GRAMMAR()

GET RESULT()

GET HYPOTHESIS()

Fig 9.

1000 ↙

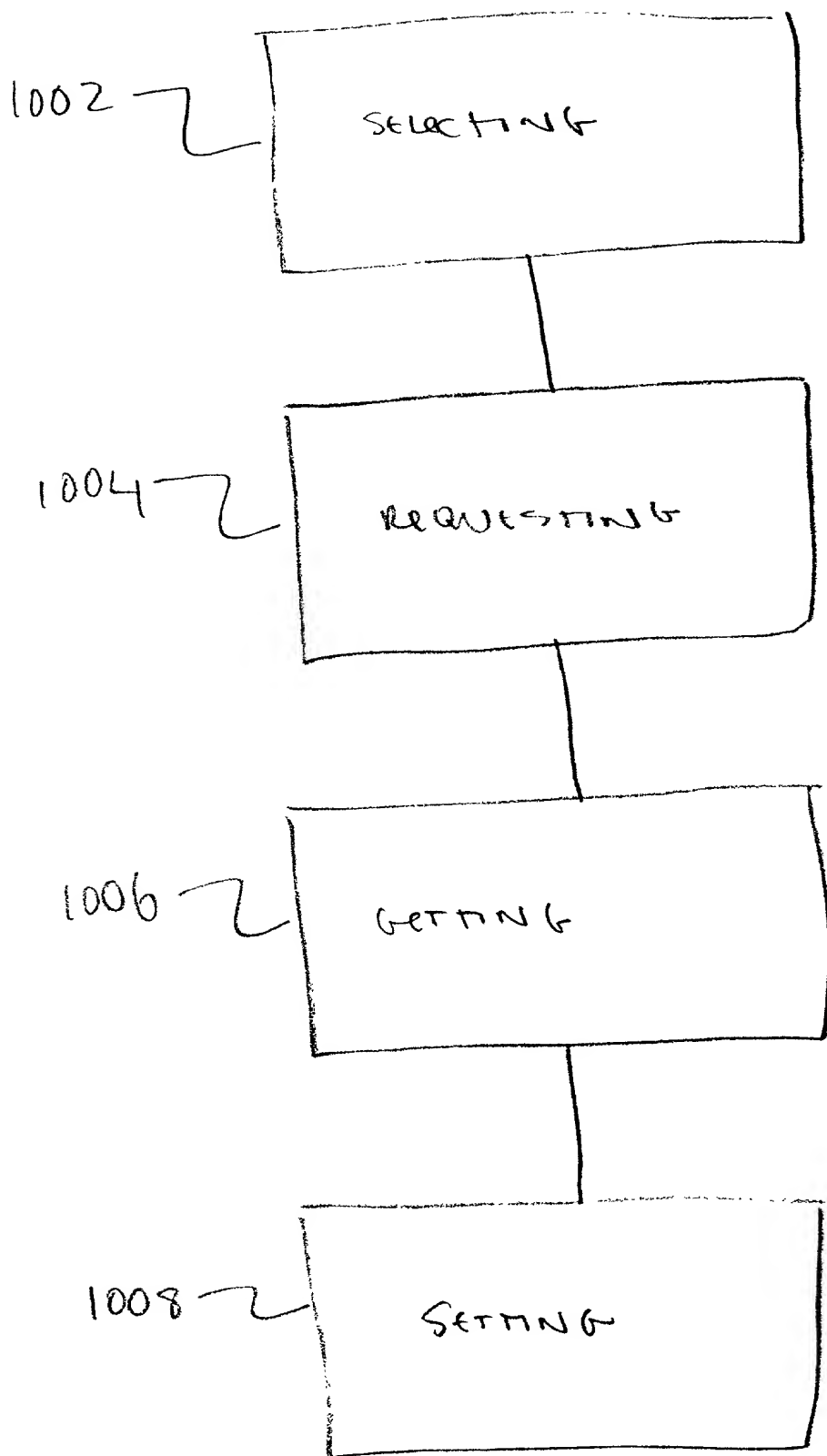


FIG. 10

# SPEECH GENERATION

SET VOICE()
GET VOICE()
SET PRIORITY()
GET PRIORITY()
SET VOLUME()
GET VOLUME()
SET RATE()
GET RATE()
SET TIME()
GET TIME()
GET STATUS()
SLEEP()
WAIT()
SPEAK()
ENUMERATE VOICES()

1102  
 1104  
 1106  
 1108  
 1110  
 1112  
 1114  
 1116  
 1118  
 1120  
 1122  
 1124  
 1126  
 1128  
 1130

1200 ↙

1202 ~

REQUESTING

1204 ~

selecting

1206 ~

GENERATING

FIG 12

SCHWEGMAN, LUNDBERG, WOESSNER &amp; KLUTH, P.A.

# United States Patent Application

## COMBINED DECLARATION AND POWER OF ATTORNEY

As a below named inventor I hereby declare that: my residence, post office address and citizenship are as stated below next to my name; that

I verily believe I am the original, first and joint inventor of the subject matter which is claimed and for which a patent is sought on the invention entitled: **SPEECH PROCESSING FOR TELEPHONY API.**

The specification of which is attached hereto.

I hereby state that I have reviewed and understand the contents of the above-identified specification, including the claims, as amended by any amendment referred to above.

I acknowledge the duty to disclose information which is material to the patentability of this application in accordance with 37 C.F.R. § 1.56 (attached hereto). I also acknowledge my duty to disclose all information known to be material to patentability which became available between a filing date of a prior application and the national or PCT international filing date in the event this is a Continuation-In-Part application in accordance with 37 C.F.R. § 1.63(e).

I hereby claim foreign priority benefits under 35 U.S.C. § 119(a)-(d) or 365(b) of any foreign application(s) for patent or inventor's certificate, or 365(a) of any PCT international application which designated at least one country other than the United States of America, listed below and have also identified below any foreign application for patent or inventor's certificate having a filing date before that of the application on the basis of which priority is claimed:

**No such claim for priority is being made at this time.**

I hereby claim the benefit under 35 U.S.C. § 119(e) of any United States provisional application(s) listed below:

**No such claim for priority is being made at this time.**

I hereby claim the benefit under 35 U.S.C. § 120 or 365(c) of any United States and PCT international application(s) listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States or PCT international application in the manner provided by the first paragraph of 35 U.S.C. § 112, I acknowledge the duty to disclose material information as defined in 37 C.F.R. § 1.56(a) which became available between the filing date of the prior application and the national or PCT international filing date of this application:

**Application Number**  
09/157,469

**Filing Date**  
September 21, 1998

**Status**  
Pending

I hereby appoint the following attorney(s) and/or patent agent(s) to prosecute this application and to transact all business in the Patent and Trademark Office connected herewith:

Anglin, J. Michael	Reg. No. 24,916	Jurkovich, Patti J.	Reg. No. 44,813	Nelson, Albin J.	Reg. No. 28,650
Bianchi, Timothy E.	Reg. No. 39,610	Kalis, Janal M.	Reg. No. 37,650	Nielsen, Walter W.	Reg. No. 25,539
Billion, Richard E.	Reg. No. 32,836	Kaufmann, John D.	Reg. No. 24,017	Oh, Allen J.	Reg. No. 42,047
Black, David W.	Reg. No. 42,331	Klima-Silberg, Catherine I.	Reg. No. 40,052	Padys, Danny J.	Reg. No. 35,635
Brennan, Leoniede M.	Reg. No. 35,832	Kluth, Daniel J.	Reg. No. 32,146	Parker, J. Kevin	Reg. No. 33,024
Brennan, Thomas F.	Reg. No. 35,075	Lacy, Rodney L.	Reg. No. 41,136	Perdok, Monique M.	Reg. No. 42,989
Brooks, Edward J., III	Reg. No. 40,925	Leffert, Thomas W.	Reg. No. 40,697	Prout, William F.	Reg. No. 33,995
Chu, Dinh C.P.	Reg. No. 41,676	Lemaire, Charles A.	Reg. No. 36,198	Sako, Katie E.	Reg. No. 32,628
Clark, Barbara J.	Reg. No. 38,107	Litman, Mark A.	Reg. No. 26,390	Schumm, Sherry W.	Reg. No. 39,422
Crouse, Daniel D.	Reg. No. 32,022	Lundberg, Steven W.	Reg. No. 30,568	Schwegman, Micheal L.	Reg. No. 25,816
Dahl, John M.	Reg. No. 44,639	Mack, Lisa K.	Reg. No. 42,825	Smith, Michael G.	Reg. No. 45,368
Drake, Eduardo E.	Reg. No. 40,594	Maeyaert, Paul L.	Reg. No. 40,076	Speier, Gary J.	Reg. No. 45,458
Eliseeva, Maria M.	Reg. No. 43,328	Maki, Peter C.	Reg. No. 42,832	Steffey, Charles E.	Reg. No. 25,179
Embreton, Janet E.	Reg. No. 39,665	Malen, Peter L.	Reg. No. 44,894	Terry, Kathleen R.	Reg. No. 31,884
Fordenbacher, Paul J.	Reg. No. 42,546	Mates, Robert E.	Reg. No. 35,271	Tong, Viet V.	Reg. No. 45,416
Forrest, Bradley A.	Reg. No. 30,837	McCrackin, Ann M.	Reg. No. 42,858	Viksnins, Ann S.	Reg. No. 37,748
Harris, Robert J.	Reg. No. 37,346	Nama, Kash	Reg. No. 44,255	Woessner, Warren D.	Reg. No. 30,440
Huebsch, Joseph C.	Reg. No. 42,673				

I hereby authorize them to act and rely on instructions from and communicate directly with the person/assignee/attorney/firm/organization/who/which first sends/sent this case to them and by whom/which I hereby declare that I have consented after full disclosure to be represented unless/until I instruct Schwegman, Lundberg, Woessner & Kluth, P.A. to the contrary.

Please direct all correspondence in this case to **Schwegman, Lundberg, Woessner & Kluth, P.A.** at the address indicated below:  
**P.O. Box 2938, Minneapolis, MN 55402**  
**Telephone No. (612)373-6900**

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

Full Name of joint inventor number 1 : **Mary Michelle Quinton**  
Citizenship: **United States of America** Residence: **Kirkland, WA**  
Post Office Address: **7012 120th Ave NE**  
**Kirkland, WA 98033**

Signature: \_\_\_\_\_ Date: \_\_\_\_\_  
Mary Michelle Quinton

Full Name of joint inventor number 2 : **Stefan Solomon**  
Citizenship: **United States of America** Residence: **Bellevue, WA**  
Post Office Address: **16827 NE 35th Street**  
**Bellevue, WA 98008**

Signature: \_\_\_\_\_ Date: \_\_\_\_\_  
Stefan Solomon

X Additional inventors are being named on separately numbered sheets, attached hereto.

Serial No. not assigned

Filing Date: not assigned

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

Full Name of joint inventor number 3 : **Donald R. Ryan**Citizenship: **United States of America**Residence: **Redmond, WA**Post Office Address: **P.O. Box 429  
Redmond, WA 98073**Signature: \_\_\_\_\_ Date: \_\_\_\_\_  
Donald R. RyanFull Name of joint inventor number 4 : **Michael Clark**Citizenship: **United States of America**Residence: **Logan, UT**Post Office Address: **940 W. 370 S.  
Logan, UT 84321**Signature: \_\_\_\_\_ Date: \_\_\_\_\_  
Michael Clark

Full Name of inventor:

Citizenship:

Residence:

Post Office Address:

Signature: \_\_\_\_\_ Date: \_\_\_\_\_

Full Name of inventor:

Citizenship:

Residence:

Post Office Address:

Signature: \_\_\_\_\_ Date: \_\_\_\_\_

§ 1.56 Duty to disclose information material to patentability.

(a) A patent by its very nature is affected with a public interest. The public interest is best served, and the most effective patent examination occurs when, at the time an application is being examined, the Office is aware of and evaluates the teachings of all information material to patentability. Each individual associated with the filing and prosecution of a patent application has a duty of candor and good faith in dealing with the Office, which includes a duty to disclose to the Office all information known to that individual to be material to patentability as defined in this section. The duty to disclose information exists with respect to each pending claim until the claim is canceled or withdrawn from consideration, or the application becomes abandoned. Information material to the patentability of a claim that is canceled or withdrawn from consideration need not be submitted if the information is not material to the patentability of any claim remaining under consideration in the application. There is no duty to submit information which is not material to the patentability of any existing claim. The duty to disclose all information known to be material to patentability is deemed to be satisfied if all information known to be material to patentability of any claim issued in a patent was cited by the Office or submitted to the Office in the manner prescribed by §§ 1.97(b)-(d) and 1.98. However, no patent will be granted on an application in connection with which fraud on the Office was practiced or attempted or the duty of disclosure was violated through bad faith or intentional misconduct. The Office encourages applicants to carefully examine:

- (1) prior art cited in search reports of a foreign patent office in a counterpart application, and
- (2) the closest information over which individuals associated with the filing or prosecution of a patent application believe any pending claim patentably defines, to make sure that any material information contained therein is disclosed to the Office.

(b) Under this section, information is material to patentability when it is not cumulative to information already of record or being made of record in the application, and

- (1) It establishes, by itself or in combination with other information, a prima facie case of unpatentability of a claim; or
- (2) It refutes, or is inconsistent with, a position the applicant takes in:
  - (i) Opposing an argument of unpatentability relied on by the Office, or
  - (ii) Asserting an argument of patentability.

A prima facie case of unpatentability is established when the information compels a conclusion that a claim is unpatentable under the preponderance of evidence, burden-of-proof standard, giving each term in the claim its broadest reasonable construction consistent with the specification, and before any consideration is given to evidence which may be submitted in an attempt to establish a contrary conclusion of patentability.

(c) Individuals associated with the filing or prosecution of a patent application within the meaning of this section are:

- (1) Each inventor named in the application;
- (2) Each attorney or agent who prepares or prosecutes the application; and
- (3) Every other person who is substantively involved in the preparation or prosecution of the application and who is associated with the inventor, with the assignee or with anyone to whom there is an obligation to assign the application.

(d) Individuals other than the attorney, agent or inventor may comply with this section by disclosing information to the attorney, agent, or inventor.